UNIVERZA V LJUBLJANI

FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Elvis Kocjan

Nadzor in pregled računalniške opreme v podjetju

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Ljubljana, 2014

UNIVERZA V LJUBLJANI

FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Elvis Kocjan

Nadzor in pregled računalniške opreme v podjetju

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: viš. pred. dr. Igor Rožanc

Ljubljana, 2014

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Učinkovit nadzor in pregled nameščene strojne in programske opreme v večjem podjetju zahteva uporabo namenskih sistemov. V diplomski nalogi preverite večje število odprtokodnih sistemov za nadzor in pregled računalniške opreme, nato pa na podlagi vgrajene opreme v podjetju, opredeljenih kriterijev in testiranja izberite najprimernejšega. V nadaljevanju prikažite delovanje sistema ter razvoj potrebne nadgradnje - izdelave vtičnika, ki omogoča uporabo sistema v skladu z varnostno politiko podjetja.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Elvis Kocjan, z vpisno številko **63980343**, sem avtor diplomskega dela z naslovom:

Nadzor in pregled računalniške opreme v podjetju

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom viš. pred. dr. Igorja Rožanca,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 15. septembra 2014 Podpis avtorja:

Za vzpodbudo, pomoč in sodelovanje pri izdelavi diplomskega dela se zahvaljujem mentorju viš. pred. dr. Igorju Rožancu in svoji ljubi ženi Andreji. Posebej se zahvaljujem svoji hčerki Maji in sinu Eneju, ki sta mi dala energijo za dokončanje študija. Njima tudi posvečam to diplomsko delo.

Svoji hčerki Maji in sinu Eneju.

Kazalo

Povzetek

Abstract

Poglavje 1 Uvod 1

Poglavje 2 Izbor odprtokodnega sistema za nadzor in pregled računalniške opreme ... 3

2.1 Posnetek stanja 3

2.2 Izbira sistema za nadzor in pregled 4

2.3 Kratek opis izbranih sistemov za nadzor 4

2.3.1 Sistem za nadzor računalniške opreme Nagios 4

2.3.2 Sistem za nadzor računalniške opreme Zabbix 5

2.3.3 Sistem za nadzor računalniške opreme Zenoss 5

2.4 Kratek opis izbranih sistemov za pregled 6

2.4.1 Sistem za pregled računalniške opreme Open-AudIT 6

2.4.2 Sistem za pregled računalniške opreme OCS Inventory NG 7

2.5 Vzpostavitev testnega okolja 7

2.6 Izbira končnega sistema 8

2.7 Integracija v produkcijsko okolje 9

Poglavje 3 Podrobnejši opis izbranih sistemov 11

3.1 Sistem za nadzor računalniške opreme Nagios 11

3.1.1 Nagios skozi zgodovino 11

3.1.2 Jedro sistema Nagios 12

3.1.3 Vtičniki za Nagios 18

3.1.4 Dodatki za Nagios 19

3.2 Sistem za pregled računalniške opreme Open-AudIT 20

3.2.1 Open-AudIT skozi zgodovino 20

3.2.2 Arhitektura sistema Open-AudIT 20
3.2.3 Vnos podatkov v sistem Open-AudIT 21
Poglavje 4 Razvoj vtičnika 23
4.1 Delovanje vtičnikov 23
4.2 Vtičnik za nadzor delovanja programskega čezmernega polja samostojnih diskov ..24
4.2.1 Posnetek stanja 24
4.2.2 Kriteriji za razvoj vtičnika 24
4.2.3 Razvoj vtičnika 25
4.2.4 Preizkus delovanja vtičnika 29
Poglavje 5 Zaključek 31
Literatura

Seznam uporabljenih kratic

kratica	angleško	slovensko
API	Application Program Interface	Programski vmesnik
AWK	Interpreted programming language designed for text processing	Programski jezik namenjen tolmačenju in obdelavi besedila
BASH	Bourne-Again Shell	Bash ukazna lupina
BSD	Berkeley Software Distribution	Berklejeva distribucija programske opreme
FTP	File Transfer Protocol	Protokol za prenos datotek
GPL	GNU General Public License	Splošna javna licenca GNU
HTTP	Hyper Text Transfer Protocol	Protokol za prenos informacij na spletu
IMAP	Internet Message Access Protocol	Protokol za dostop do e-pošte na oddaljenem strežniku
JMX	Java Management Extensions	Razširitve za upravljanje Jave
IP	Internet protocol	Internetni protokol
IPMI	Intelligent Platform Management Interface	Inteligentni vmesnik za upravljanje računalniškega okolja
NAS	Network Attached Storage	Omrežno diskovno polje
NMAP	Network Mapper	Načrtovalec omrežij
RAID	Redundant array of independent disks	Čezmerno polje samostojnih diskov
SHELL	Command Line Interpreter	Tolmač ukaznih vrstic
SMS	Short Message Service	Storitev za pošiljanje kratkih sporočil
SMTP	Simple Mail Transfer Protocol	Preprosti protokol za prenos pošte

SNMP	Simple Network Management Protocol	Preprosti protokol za upravljanje omrežij
SQL	Structured Query Language	Sestavljeni jezik za poizvedbe
SSH	Secure Shell (cryptographic network protocol for secure data communication)	Varna lupina (komunikacijski protokol za varno povezavo med dvema računalnikoma)
UPS	Uninterruptible Power Supply	Brezprekinitveno napajanje
VPN	Virtual Private Network	Navidezno zasebno omrežje
XMPP	Extensible Messaging and Presence Protocol	Zbirka odprtih protokolov in tehnologij

Povzetek

Diplomska naloga opisuje izbiro odprtokodnega sistema za nadzor in pregled računalniške opreme v podjetju, ki si komercialne programske opreme za ta namen ne more privoščiti.

Na kratko smo predstavili sisteme za nadzor (Nagios, Zabbix in Zenoss) in pregled (Open-AudIT in OCS Inventory NG) računalniške opreme. Na podlagi kriterijev in lastnega testiranja smo izmed predstavljenih izbrali Nagios kot sistem za nadzor in Open-AudIT kot sistem za pregled. V nadaljevanju smo podrobneje opisali izbrana sistema, kjer smo zajeli bistvene značilnosti za razumevanje njunega delovanja. Na koncu smo prikazali še razvoj vtičnika, ki smo ga razvili za potrebe nadzora namenske naprave v podjetju.

Na uradni strani Nagiosa najdemo veliko število vtičnikov, tako uradnih s strani avtorja sistema kot neuradnih, ki so jih razvili uporabniki sistema. Zaradi varnostne politike v podjetju nam obstoječi vtičniki niso koristili, kar je bil razlog za razvoj lastnega vtičnika.

Odločitev za uvedbo sistema za nadzor in pregled se je pokazala kot dobra in zanesljiva, saj po uvedbi sistemov odprava napak na omrežnih napravah in storitvah poteka bistveno hitreje, nekatere napake pa se da tudi predvideti.

Ključne besede: nadzor računalniške opreme, pregled računalniške opreme, Nagios, Open-AudIT, vtičnik, čezmerno polje samostojnih diskov.

Abstract

The thesis describes selection of an open source monitoring and auditing system for computer equipment in an enterprise, where the cost of same purpose commercial software is too high.

Control systems (Nagios, Zabbix and Zenoss) and auditing systems (Open-Audit and OCS Inventory NG) for computer equipment are shortly presented first. Using the comparative criteria and our own testing we selected Nagios as the most suitable monitoring system and Open-Audit as the best auditing system in our case. Additionally both selected systems are described in detail, including the essential features of their functionality. Finally, we presented the development of a specific plugin for the control of a special device in the enterprise.

A large number of plugins exist on the official site of Nagios, some written by the author of the Nagios system and others developed by other system users. Due to security policy of the enterprise existing plugins were not beneficial, thus we developed our own plugin.

The decision to establish a monitoring and auditing system proved to be correct and reliable. Since its introduction error correction on the network devices and services runs significantly faster, and some errors can be foreseen as well.

Keywords: computer equipment monitoring, computer equipment auditing, Nagios, Open-AudIT, plugin, redundant array of independent disks.

Poglavje 1

Uvod

Računalniška omrežja in z omrežji povezane naprave so se začele pojavljati v sedemdesetih letih dvajsetega stoletja. Od začetka do danes so z razvojem tehnologij postale del našega vsakdana tako v podjetju kot doma. Lahko bi rekli, da računalniška omrežja in naprave danes v podjetju omogočajo posredovanje in hrambo poslovnih podatkov.

Uspešnost podjetja je v veliki meri odvisna od informacijskih in komunikacijskih tehnologij, zato se z razvojem teh tehnologij veča število omrežnih naprav in posledično širi omrežje podjetja. Tako postaja podjetje vse bolj odvisno od informacijskih in komunikacijskih tehnologij, to pa pomeni, da je potrebno nadzoru delovanja in pregledu naprav dati velik pomen.

Za ta problem najdemo na tržišču tako komercialne, kot odprtokodne rešitve. Odločili smo se, da v diplomski nalogi preizkusimo odprtokodne rešitve, saj nam te poleg nizkih stroškov omogočajo še prilagoditev lastnim potrebam in željam. Velik pomen pri odločitvi igra tudi finančni vložek podjetja v sistem.

V diplomski nalogi je opisan izbor odprtokodnega sistema za nadzor in pregled računalniške opreme v podjetju. V ta namen smo poiskali pet sistemov (Nagios, Zabbix, Zenoss, Open-AudIT, OCS Inventory NG), ter se na podlagi testiranja odločili za Nagios in Open-AudIT. Pri tem smo si pomagali s primerjalnimi kriteriji. V nadaljevanju smo oba sistema podrobneje predstavili. Da je sistem za nadzor opravljal svojo funkcijo, kot smo si zaželeli, smo razvili tudi Nagios vtičnik za nadzor delovanja programskega čezmernega polja samostojnih diskov (RAID). Naloga je zaključena s preizkusom delovanja razvitega vtičnika v testnem okolju.

Poglavje 2

Izbor odprtokodnega sistema za nadzor in pregled računalniške opreme

2.1 Posnetek stanja

V podjetju na oddelku za računalništvo in informatiko skrbimo za preko 160 omrežnih naprav in preko 80 namenskih naprav. Med omrežne naprave štejemo vse naprave, ki imajo omrežni dostop. Te naprave so strežniki, delovne postaje, omrežni tiskalniki, usmerjevalniki in stikala. Med namenske naprave štejemo tiskalnike brez omrežne podpore, optične bralnike, večnamenske naprave (tiskalnik, optični bralnik in faks v eni napravi), brezprekinitvena napajanja (ang. `uninterruptible power suply`), projektorje, itd.

Na strežnikih in delovnih postajah imamo nameščene različne operacijske sisteme, večinoma so to Microsoft Windows in Debian GNU/Linux, v testnem virtualnem okolju pa imamo še CentOS GNU/Linux, Zentyal [1], OpenBSD, Oracle Solaris in ReactOS [2]. Na strežnikih, ki temeljijo na Microsoft Windows računalniškem okolju (ang. `platform`), imamo nameščen Microsoft SQL strežnik in aplikacijski strežnik, na računalniškem okolju GNU/Linux pa imamo nameščen poštni strežnik, FTP strežnik, NAS strežnik, MySql strežnik in Samba [3] datotečni strežnik. Poudariti je treba, da skrbimo še za nameščanje, posodabljanje in delovanje programske opreme, ki teče tako na omrežnih, kot na namenskih napravah. Med programsko opremo štejemo operacijske sisteme, strežniško programsko opremo, namensko programsko opremo (Microsoft Office, LibreOffice, Autodesk AutoCad, itd.) in strojno programsko opremo (ang. `firmware`).

Slab nadzor in pregled zaradi raznolikosti omrežnih in namenskih naprav ter programske opreme je bil povod, da smo v podjetju začeli razmišljali o uvedbi sistema. Z uvedbo takega sistema smo želeli in pričakovali, da bo v veliki meri razbremenil zaposlene, ki skrbimo za nemoteno delovanje in upravljanje s temi napravami.

2.2 Izbira sistema za nadzor in pregled

Na tržišču obstaja veliko sistemov za nadzor in pregled računalniške opreme. Zaradi tega je pomembno, da smo pri izbiri pazljivi in si definiramo kriterije. S pomočjo kriterijev smo primerjali sisteme, ki bi lahko zadovoljili potrebe našega podjetja.

Odločitev o tem, katere sisteme bomo testirali, smo sprejeli na podlagi strokovnih člankov s svetovnega spleta [4] ter priporočil s forumov [5] in seminarjev [6][7]. Veliko člankov je vidno opredeljenih za določeno rešitev, zato smo pri tem morali biti še posebej previdni. Bistvena kriterija sta bila odprtokodnost sistema, saj smo si želeli možnosti dograjevanja sistema ter brezplačnost, saj bi ga namestili, nastavili in upravljali sami. Drug kriterij pa je bila zasnova sistema na spletnem strežniku, predvsem zaradi lažjega in hitrejšega dostopanja do podatkov.

Nagios, Zabbix in Zenoss so sistemi za nadzor, za katere smo presodili, da bi lahko zadovoljili potrebe podjetja po nadzoru. Naša želja je bila, da bi sistem omogočal tudi pregled računalniške opreme, vendar na tržišču zadovoljive rešitve nismo našli. Open-AudIT in OCS Inventory NG sta bila primerna kandidata za ta del.

2.3 Kratek opis izbranih sistemov za nadzor

Namen takega sistema je, da omogoča nadzor in analizo lokalnih in omrežnih storitev, ki tečejo na različnih napravah ves čas njihovega delovanja. Na podlagi informacij, ki nam jih sistem vrne, dobimo strukturiran pregled delovanja celotnega omrežja.

2.3.1 Sistem za nadzor računalniške opreme Nagios

Nagios [8] se je že od samega začetka razvijal za delovanje pod odprtokodnim operacijskim sistemom GNU/Linux, deluje pa tudi pod ostalimi UNIX sorodnimi operacijskimi sistemi (BSD, Oracle Solaris, OS X, Minix, itd.). Jedro sistema je napisano v programskem jeziku C, spletni vmesnik pa v skriptnem programskem jeziku PHP.

Bistvene možnosti nadzora, ki ga Nagios omogoča:

- nadzor strežnikov in delovnih postaj (obremenitev procesorja, izkoriščenost trdega diska in delovnega pomnilnika, itd.),
- nadzor različnih omrežnih storitev (SMTP, IMAP, http, SQL baza, PING, itd.),
- nadzor omrežnih naprav (stikala, usmerjevalniki, tiskalniki, itd.),

- nadzor lokalnega omrežja in povezav v svetovni splet (obremenjenost, odzivnost, razpoložljivost),
- vzporedni nadzor s časovnim razmikom, ki ga določimo sami,
- obveščanje uporabnika v primeru napak (preko SMS sporočila, e-pošte, itd.), itd.

2.3.2 Sistem za nadzor računalniške opreme Zabbix

Zabbix [9] deluje na vseh UNIX sorodnih operacijskih sistemih. Ravno tako kot pri Nagiosu je njegovo jedro napisano v programskem jeziku C in spletni vmesnik v skriptnem programskem jeziku PHP.

Bistvene možnosti nadzora, ki ga Zabbix omogoča:

- nadzor različnih omrežnih storitev (SMTP, IMAP, HTTP, itd.) preko t.i. preprostega pregleda,
- nadzor strežnikov in delovnih postaj preko prednameščenih agentov (obremenitev procesorja, izkoriščenost trdega diska in delovnega pomnilnika, itd.),
- nadzor omrežnih naprav preko protokolov, ki omogočajo pregled in upravljanje (SNMP, SSH, IPMI [10], JMX [11], itd.),
- obveščanje uporabnika v primeru napak (XMPP [12], e-pošta, SMS sporočila), itd.

2.3.3 Sistem za nadzor računalniške opreme Zenoss

Zenoss [13] deluje na vseh UNIX sorodnih operacijskih sistemih, toda uradno sta podprta Red Hat Enterprise in CentOS GNU/Linux. Neuradno sta podprta tudi Debian in Ubuntu GNU/Linux. Na spletni strani je možno dobiti virtualno sliko prednameščenega celotnega sistema. Za razliko od Nagiosa in Zabbixa, je Zenoss napisan v skriptnem programskem jeziku Python in programskem jeziku Java. Spletni vmesnik je razvit na podlagi spletnega aplikacijskega ogrodja Zope [14].

Bistvene možnosti nadzora, ki ga Zenoss omogoča:

- nadzor različnih omrežnih storitev (SMTP, IMAP, HTTP, itd.),
- nadzor omrežnih naprav (obremenitev procesorja, izkoriščenost trdega diska in delovnega pomnilnika, itd.) preko vtičnikov ali dodatkov,

- orodja za upravljanje dogodkov na podlagi sistemskih opozoril,
- samodejno iskanje omrežnih naprav in sprememb v konfiguraciji omrežja,
- podpora za Nagios vtičnike, itd.

2.4 Kratek opis izbranih sistemov za pregled

Sistemi za pregled so namenjeni pregledu omrežnih naprav, pregledu nastavitev teh naprav in pregledu vseh sprememb, ki se z omrežnimi napravami lahko zgodijo. Poleg tega omogočajo tudi pregled vse programske in strojne opreme, ki je nameščena na omrežnih napravah.

Za razliko od nadzornih sistemov, sistema za pregled delujeta tako pod GNU/Linux kot Windows operacijskimi sistemi.

2.4.1 Sistem za pregled računalniške opreme Open-AudIT

Za Open-AudIT [15] lahko rečemo, da je podatkovna baza podatkov, ki se polni na Windows sistemih z VBScript (`Visual Basic Scripting Edition`) ukaznimi datotekami. Na GNU/Linux sistemih pa z Bash ukaznimi datotekami ter ostalimi namenskimi ukaznimi datotekami. Spletni vmesnik je razvit v skriptnem programskem jeziku PHP.

Sistem za pregled Open-AudIT nam omogoča:

- pregledovanje omrežne opreme iz domenskega strežnika (Microsoft Active Directory),
- pregledovanje delovnih postaj in strežnikov, na katerih so nameščeni GNU/Linux, Windows ali OSX operacijski sistemi,
- pregledovanje omrežnih naprav (tiskalniki, stikala, usmerjevalniki, itd.),
- podomrežno zajemanje (ang. `subnet scanning`) s programsko opremo `NMAP` [16] in pregledovanje zajete omrežne opreme,
- ročni vnos naprav preko preglednic.

Open-AudIT ima možnost, da zajema omrežja in omrežne naprave avtomatično. Ta zajem je priporočljivo izvajati vsakih nekaj ur, saj imamo tako boljši pregled nad spremembami v omrežju.

2.4.2 Sistem za pregled računalniške opreme OCS Inventory NG

OCS Inventory NG [17] (`Open Computer and Software Inventory Next Generation`) pridobiva podatke s pomočjo prednameščenega klienta (`OCS Inventory NG Agent` [18]). Klienta lahko namestimo na UNIX sorodne in Microsoft Windows operacijske sisteme. Ostale naprave (stikala, usmerjevalniki, tiskalniki, itd.) pa pregleduje s pomočjo agentov ki pregledujejo podomrežje. Sistem je razvit s skriptnima programskima jezikoma Perl in PHP.

Sistem za pregled OCS Inventory NG nam omogoča:

- nameščanje in posodabljanje programske opreme na omrežnih napravah, kjer je nameščen klient,
- podomrežno zajemanje (ang. `subnet scanning`) s pomočjo agentov,
- ročni vnos naprav preko namenskih ukaznih datotek,
- možnost integracije lastnih vtičnikov, itd.

2.5 Vzpostavitev testnega okolja

Testiranje je potekalo v virtualnem okolju na obstoječi strojni in programski opremi. Virtualno okolje nam omogoča, da s pomočjo informacijskih tehnologij vzpostavimo simulirano okolje, ki daje vtis resničnega in ima funkcionalnosti stvarnega okolja. V ta namen smo uporabili programsko opremo `VMWare Workstation 10` [19], nameščeno na močnejšem osebnem računalniku (procesor Intel Core i5, bralno-pisalni spomin 16GB, 1,5TB trdi disk) z operacijskim sistemom Debian GNU/Linux 7 64bit. Na tej opremi smo vzpostavili virtualne strežnike za vsak sistem za nadzor in pregled.

Pomembnejša konfiguracija virtualnih strežnikov:

- virtualna strojna oprema: 4 virtualni procesorji, 2GB bralno-pisalnega spomina in 20 GB virtualnega trdega diska,
- operacijski sistem: Debian GNU/Linux 7 Wheezy 64bit,
- podatkovna baza: MySQL 5 in
- spletni strežnik: Apache in Zope [14] (Zenoss).

Uporabili smo dve delovni postaji, ki sta bili namenjeni nadzoru in pregledu. Ti napravi sta bili namenjeni za del testa, kjer so sistemi za nadzor in pregled potrebovali namestitev klientov. Na napravah je bil nameščen operacijski sistem Microsoft Windows 7 in Debian GNU/Linux. V primeru testov brez nameščenega klienta, smo uporabili obstoječo opremo, saj s tem nismo vplivali na konfiguracijo in delovanje naprave. Vse naprave so bile povezane v lokalno omrežje.

2.6 Izbira končnega sistema

Za primerjavo sistemov smo uporabili naslednje primerjalne kriterije z ocenami od ena do tri:

- zahteve:
  - sistemske zahteve (1 - visoke, 2 - srednje, 3 - nizke),
  - odzivnost sistema(1- nezadovoljiva, 2 - zadovoljiva, 3 - dobra),
- namestitev:
  - potrebno znanje za izvedbo namestitve (1 - odlično, 2 - dobro, 3 - zadovoljivo),
  - pomoč in podpora (1 - slaba, 2 - srednja, 3 - dobra),
- uporabnost:
  - preglednost spletnega vmesnika (1- nezadovoljiva, 2 - zadovoljiva, 3 - dobra),
  - odzivnost spletnega vmesnika (1- nezadovoljiva, 2 - zadovoljiva, 3 - dobra),
- prilagoditev:
  - težavnost nastavitve sistema (1 - visoka, 2 - srednja, 3 - nizka),
  - potrebno znanje za dograjevanje sistema (1 - visoka, 2 - srednja, 3 - nizka).

Rezultati našega testa so prikazani v tabeli (tabela 2.1).

Sistem za nadzor in pregled	Primerjalni kriteriji							
	Zahteve		Namestitev		Uporabnost		Prilagoditev	
	Sistemske zahteve	Odzivnost sistema	Potrebno znanje za namestitev	Pomoč in podpora	Preglednost spletnega vmesnika	Odzivnost spletnega vmesnika	Težavnost nastavitve sistema	Potrebno znanje za dograjevanje sistema
Nagios	2	3	3	3	3	3	2	3
Zabbix	2	3	3	2	2	3	3	2
Zenoss	2	2	2	1	2	1	2	2
Open-AudIT	3	3	2	2	3	3	2	1
OCS Inventory NG	3	2	2	2	1	2	2	1

Tabela 2.1: Rezultati primerjave sistemov po primerjalnih kriterijih

Na podlagi končne ocene testiranja (tabela 2.2) smo se odločili, da izberemo sistem za nadzor Nagios in sistem za pregled Open-AudIT.

	Sistemi za nadzor			Sistemi za pregled	
	Nagios	Zabbix	Zenoss	Open-AudIT	OCS Inventory NG
Končna ocena	22	20	14	19	15

Tabela 2.2: Končna ocena testiranja

2.7 Integracija v produkcijsko okolje

Po končanem testiranju smo oba izbrana sistema integrirali v obstoječe produkcijsko okolje. Pri vzpostavitvi testnega okolja smo bili še posebej previdni, saj smo se odločili, da bo sistem, izbran na podlagi primerjalnih kriterijev, ostal in ga ne bomo ponovno vzpostavljali.

Potrebno je poudariti, da del produkcijskega okolja v podjetju temelji na virtualizaciji in smo virtualna strežnika iz testnega okolja prenesli v produkcijsko okolje. To nam omogoča programska oprema `VMWare Workstation 10`, saj v produkcijskem okolju uporabljamo programsko opremo `vSphere Hypervisor` [20], nameščeno na `HP ProLiant BL460c` strežniški rezini. Programski opremi sta od istega proizvajalca `VMware` in sta medsebojno kompatibilni.

Izbrana sistema tako Nagios kot Open-Audit sta zadovoljila vsa pričakovanja saj se je bistveno izboljšal nadzor in pregled računalniške opreme. Kadar želimo recimo preveriti stanje določene storitve ali naprave, uporabimo Nagios, če pa želimo recimo vedeti, kakšna programska oprema je nameščena na določeni delovni postaji, pa uporabimo Open-AudIT.

Poglavje 3

Podrobnejši opis izbranih sistemov

3.1 Sistem za nadzor računalniške opreme Nagios

Za delovanje Nagiosa [8] so ključni vtičniki, saj je brez njih jedro sistema, ki od vtičnikov sprejema signale tako, rekoč neuporabno. Z namestitvijo paketa `Nagios Plugins` [21] dobimo nekaj osnovnih vtičnikov, imamo pa tudi možnost, da te vtičnike prilagodimo svojim potrebam, ali pa jih preprosto sami napišemo za želeno storitev ali napravo. Veliko število vtičnikov ponuja tudi odprtokodna skupnost `Nagios Plugins`.

Princip delovanja Nagiosa bi na kratko lahko opisali na naslednji način: ko se zgodi zahteva za preverjanje stanja določene nadzorovane storitve ali naprave, Nagiosovo jedro pokliče vtičnik. Ta pokliče oddaljeno ali lokalno storitev, preveri njen odziv in jedru vrne rezultat želene poizvedbe.

Prejete rezultate oz. informacije jedro sistema obdela in shrani v datotečni sistem. Če želimo podatke shranjevati v podatkovno bazo, si moramo predhodno namestiti dodatek (ang. `addon`), ki to omogoča. Po obdelavi podatkov lahko sprožimo upravitelje dogodkov (ang. `event handlers`) zapisane v nastavitveni datoteki in z njimi odpravimo nezaželene napake, ali obvestimo uporabnika sistema.

3.1.1 Nagios skozi zgodovino

Začetki razvoja Nagiosa segajo v leto 1996, ko je avtor sistema Ethan Galstad napisal preprost program za operacijski sistem MS-DOS, ki je s `ping` ukazom sporočil `Novell Netware` strežnikom, naj mu ti s pomočjo drugih programov (vtičnikov), ki so tekli na teh strežnikih, vrnejo želene podatke. Tako je bil tudi rojen osnovni arhitekturni koncept, ki je bil uporabljen najprej za `NetSaint` [22] kasneje Nagios in leto kasneje ideja o sistemu za nadzor računalniške opreme.

Leta 1999 je avtor ocenil, da se s podobnimi težavami spopadajo ostali skrbniki računalniških omrežij in se odločil svoje delo objaviti kot odprtokodna projekta z imenom NetSaint in Natsaint Plugins v skladu z licenco GNU GPL verzija 2.

Da bi se izognil morebitnim pravnim težavam glede podobnosti imena blagovne znamke, se je Ethan leta 2002 odločil, da projekta preimenuje v Nagios in Nagios Plugins. Ime Nagios je sestavljeno iz grške besede agios, ki pomeni svetnik (ang. saint) ter črke n, ki je prva črka besede mreža (ang. network), lahko bi rekli, da je avtor staro ime delno prevedel v grški jezik in tako obdržal isti pomen besede.

Projekt je od nastanka pa do danes prejel veliko nagrad in velja za enega izmed najboljših in najbolj razširjenih odprtokodnih sistemov za nadzor računalniške opreme. Trenutno šteje že verzijo 4.0.8.

3.1.2 Jedro sistema Nagios

Bistvo Nagiosa je njegovo jedro. Njegova naloga je, da nadzoruje status storitev, ki tečejo bodisi lokalno ali oddaljeno na strežnikih, delovnih postajah ali omrežnih napravah. V primeru, ko odpove določena oprema na omrežju (status DOWN), bodo z njo odpovedale tudi vse storitve, ki so od te naprave odvisne. Lahko se tudi zgodi, da postanejo naprave v omrežju nedostopne zaradi nedelovanja naprav in povezav, ki skrbijo za komunikacijo med njimi in Nagios strežnikom. Nagios v tem primeru ne bo mogel nadzorovati storitev, ki pripadajo tem napravam (status UNREACHABLE).

Jedro Nagiosa je zasnovano tako, da prepozna takšno situacijo. Ko se pojavijo težave pri nadzoru storitev, preveri delovanje po naslednjem scenariju:

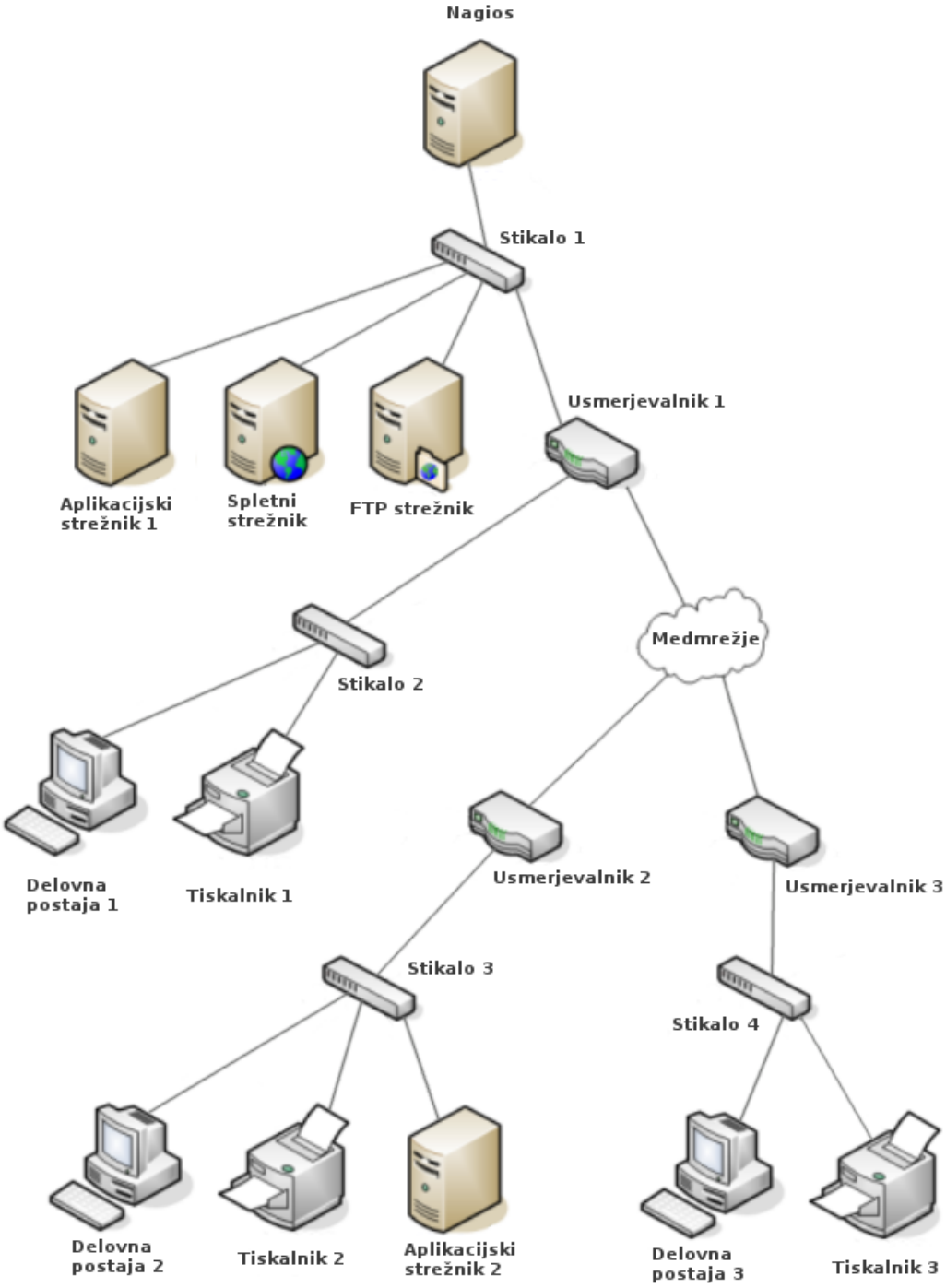
1. Preveri status storitve in če mu ta vrne Non-OK status, bo preveril še, če napravo zazna v omrežju. Običajno to stori z ukazom ping in čaka na odgovor.

2. V primeru, da zopet dobi Non-OK status, predvideva, da je z napravo v omrežju nekaj narobe, zato zaustavi vse poizvedbe storitev in sporoči uporabniku sistema, da je naprava izklopljena ali nedosegljiva.

3. V nasprotnem primeru (ko dobi OK status), pa bo Nagios zaključil, da nadzorovana storitev ne deluje pravilno in to prav tako sporoči uporabniku sistema.

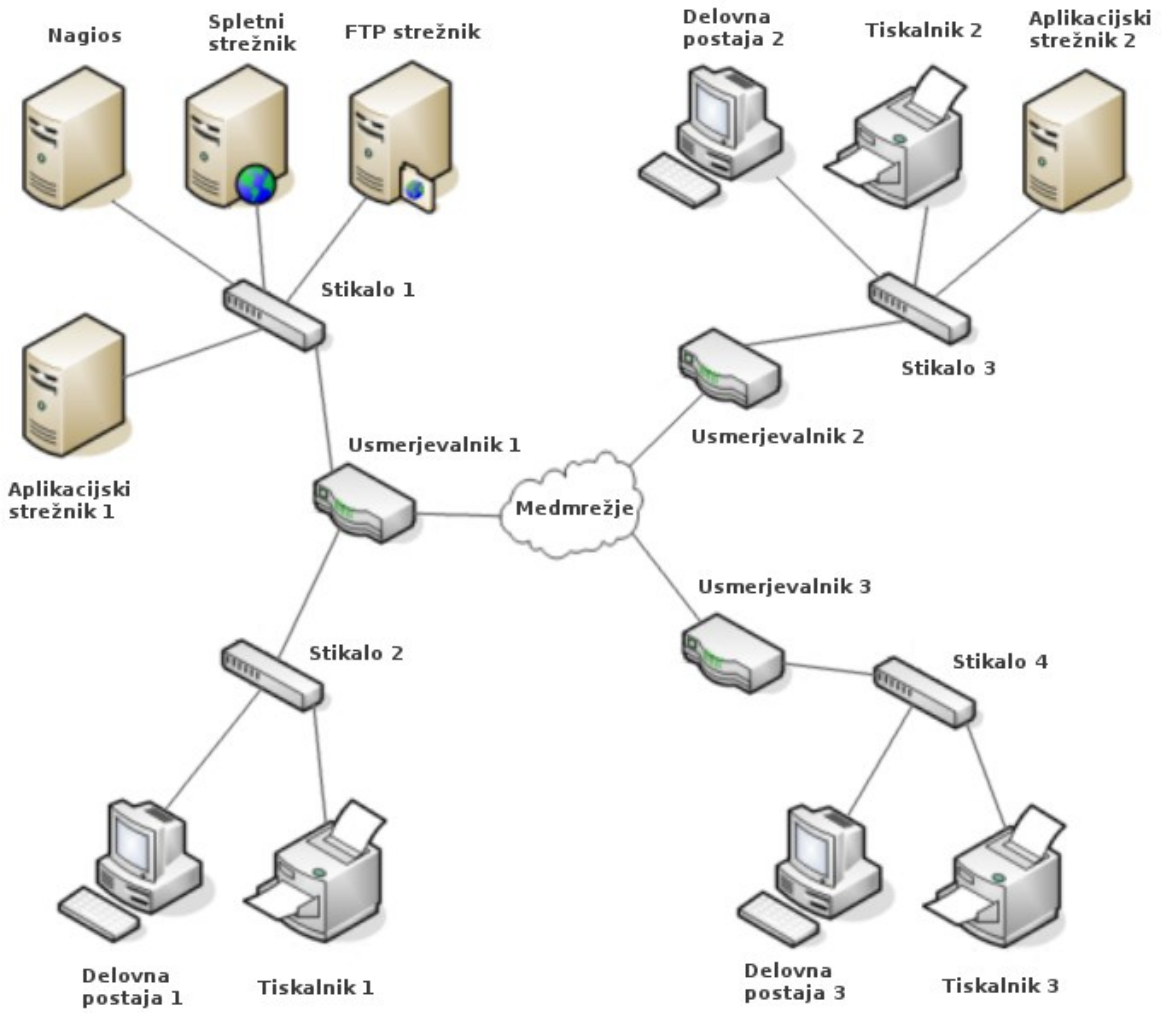
Opisana situacija deluje le v primeru, če smo v konfiguraciji jasno določili hierarhijo omrežnih naprav v odnosu z Nagiosom (razmerja starši-otroci).

Gostitelj Nagiosa je vedno na vrhu hierarhije nadzorovanih naprav, kar se lepo vidi na sliki 3.1. Ostale naprave so zanj `lokalne` ali `oddaljene`. Naprave na istem omrežnem segmentu Nagios vidi kot `lokalne naprave`, ker med njimi ni usmerjevalnikov ali požarnih zidov. Ostale naprave so za Nagios `oddaljene naprave`. Na sliki 3.2 je podan primer takih naprav v našem podjetju.



Slika 3.1: Hierarhija naprav iz vidika nadzora v našem podjetju

Na sliki 3.1 se lepo vidi hierarhija v našem podjetju. V tem primeru je Stikalo 2 otrok očeta Usmerjevalnik 1 ali pa, da je Stikalo 3 oče otrok delovne postaje 2, tiskalnika 2 in aplikacijskega strežnika 2. Stikalo 1 nima starša, ker je na istem mrežnem segmentu kot nadzorni sistem Nagios.



Slika 3.2: Primer lokalnih in oddaljenih naprav v našem podjetju

Trenutno stanje naprav v omrežju in njihovih storitvah je določeno z dvema komponentama: s statusom in stanjem. Stanje se nadalje deli še na stanje omrežnih naprav in stanje omrežnih storitev.

Nagios pozna tri vrste statusov omrežnih naprav:

- `UP` - naprava deluje,
- `DOWN` - storitev ali naprava ne deluje,
- `UNREACHABLE` - storitev ali naprava ni dosegljiva.

Omrežne storitve pa so lahko v enemu izmed statusov:

- OK - storitev deluje na pričakovan način,
- WARNING - storitev deluje, ampak je presegla določene kriterije,
- CRITICAL - storitev ne deluje pravilno, ali sploh ne deluje,
- UNKNOWN - gre za nedoločeno storitev, ki se je iz neznanih razlogov ni dalo ovrednotiti.

Tako omrežne storitve kot omrežne naprave se lahko pojavijo v t.i. lažjem stanju (SOFT state) ali t.i. težjem stanju (HARD state).

Omrežne naprave in storitve so v lažjem stanju:

- ko preverjanje statusa naprave ali storitve vrne non-OK ali non-UP status, vendar še nismo dosegli števila ponovnih preverjanj, ki jih nastavimo v konfiguraciji, temu rečemo stanje lažje napake (ang. soft error state),
- ko si naprava ali storitev iz stanja lažje napake opomoreta. Temu rečemo lažje okrevanje (ang. soft recovery).

Omrežne naprave in storitve so v težjem stanju:

- ko preverjanje statusa naprave ali storitve vrne non-OK ali non-UP stanje in smo dosegli število ponovnih preverjanj, temu stanju rečemo stanje težje napake (ang. hard error state),
- ko naprava ali storitev prehaja iz enega stanja težje napake v drugo stanje težje napake (iz statusa WARNING v status CRITICAL),
- ko preverjanje storitve vrne non-OK stanje in je gostitelj storitve v statusu DOWN ali UNREACHABLE,
- ko si iz stanja težje napake opomore. Temu rečemo težje okrevanje (ang. hard recovery).

V obeh primerih se težave z omrežnimi storitvami in omrežnimi napravami zabeležijo in izvedejo se upravitelji dogodkov. V primeru težje napake pa se o tem obvesti uporabnike sistema Nagios.

Upravitelji dogodkov so izbirne sistemske ukazne datoteke, ki se izvedejo kadarkoli se zgodi sprememba stanja omrežne naprave ali storitve. Njihova naloga je, da poizkusijo rešiti težavo, še preden se obvesti uporabnika o tej napaki.

Najpogosteje se upravitelje dogodkov uporablja:

- za ponovni zagon nedelujočih ali napačno delujočih storitev,
- za ponovni zagon omrežnih naprav,
- v primeru uporabe sistema za prijavljanje napak lahko napake prijavimo preko upraviteljev dogodkov,
- za beleženje delovanja dogodkov v podatkovno bazo.

Upravitelji dogodkov se izvedejo v primeru, ko:

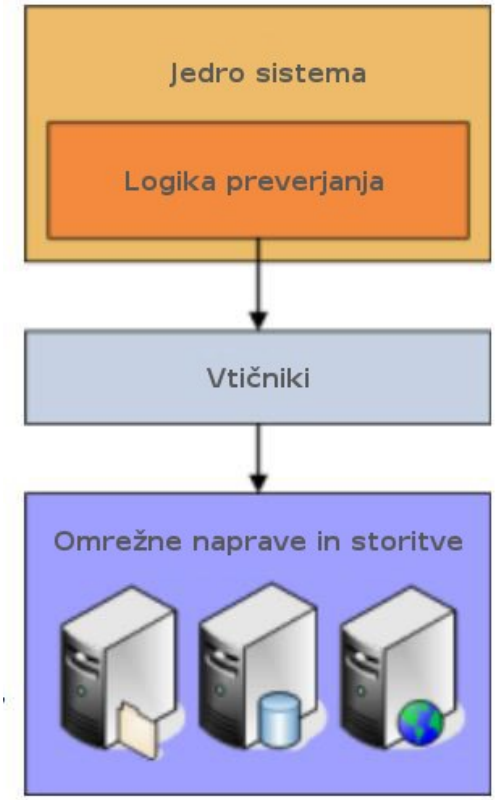
- se omrežna naprava ali storitev pojavi v stanju lažje napake,
- omrežna naprava ali storitev prvikrat preide v stanje težje napake,
- si omrežna naprava ali storitev opomore iz lažje in težje napake.

Zelo pomembna funkcija jedra Nagios je tudi obveščanje in alarmiranje uporabnika. Nagios sprejme odločitev o pošiljanju obvestila na podlagi preverjanja delovanja omrežnih naprav in storitev. S pomočjo nastavitvene datoteke, nam Nagios omogoča, da si določene zadeve uredimo po svoje.

Pošiljanje obvestila se zgodi v primeru, ko:

- se omrežne naprave ali storitve nahajajo v stanju težje napake,
- omrežne naprave ali storitve ostanejo v stanju težje napake in preverjanje naprave ali storitve vrne `non-OK` status.

Nadzor omrežnih naprav in storitev poteka v Nagiosu po aktivni ali pasivni metodi, njihove poizvedbe pa sproža logika preverjanja. Pogostejša je aktivna metoda. Ko pride do zahteve po nadzoru omrežne naprave ali storitve, logika preverjanja sporoči vtičniku kaj jo zanima in ga zažene. Vtičnik preveri želene podatke, jih vrne jedru sistema, ta pa na podlagi teh podatkov sproži morebitne akcije, odvisne od statusa omrežne naprave ali storitve. Primer takih poizvedb so recimo javno dostopne omrežne naprave ali storitve (dostopnost e-poštnega strežnika, FTP strežnika, spletnega strežnika, itd.). Delovanje aktivne poizvedbe lahko vidimo na sliki 3.3.



Slika 3.3: Aktivne poizvedbe omrežnih naprav ali storitev [8]

Pasivna metoda nadzora pride v poštev takrat, kadar se omrežne naprave nahajajo za požarnimi zidovi, imajo nameščene požarne zidove ali če poizvedbe izvajajo zunanje aplikacije (vtičniki, SNMP pasti, itd.). Delovanje pasivne metode nadzora bi lahko opisali na naslednji način: ko zunanja aplikacija sproži poizvedbo storitve, se podatki poizvedbe prenesejo na Nagios, kjer preko logike preverjanja zunanjih podatkov čakajo na nadaljnjo obdelavo. Delovanje je prikazano na sliki 3.4.

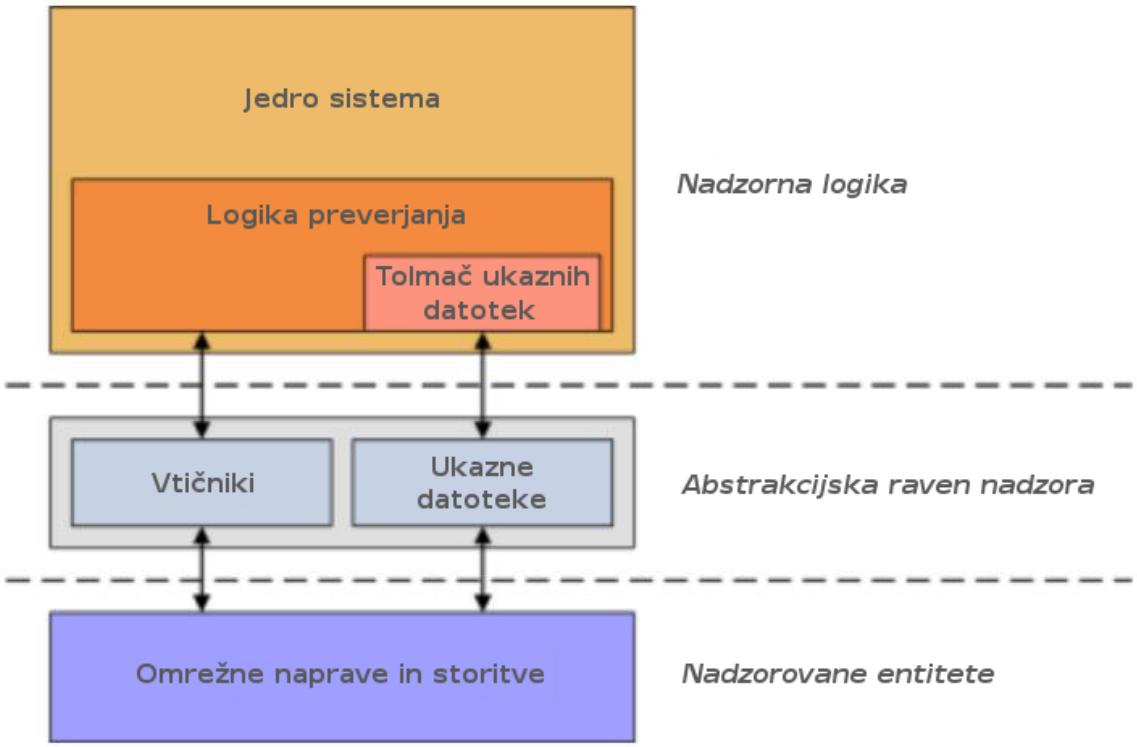
V tem poglavju so opisani poglavitni deli jedra sistema Nagios, ki so potrebni za razumevanje njegovega delovanja in osnovne konfiguracije. Podrobnejša dokumentacija je na voljo na [8].

Slika 3.4: Pasivne poizvedbe omrežnih naprav ali storitev [8]

3.1.3 Vtičniki za Nagios

Jedro Nagiosa ne vsebuje nobenega internega mehanizma za preverjanje stanja omrežnih naprav in storitev. To delo opravljajo vtičniki, ki so zunanji programi ali ukazne datoteke. Vtičniki predstavljajo izvedbeni del nadzornega sistema, zato je jedro Nagiosa brez njih neuporabno. Delujejo kot abstrakcijska raven nadzora med nadzorno logiko in omrežnimi napravami ter storitvami, ki jih nadziramo. To se lepo vidi na sliki 3.5.

Vtičnike lahko pišemo v vseh programskih jezikih, ki jih podpira nadzirana naprava. Večina vtičnikov je napisana v programskem jeziku `C` in `C++`, saj so ti vtičniki hitri in primerni za nadzor ogromnega števila omrežnih naprav in storitev. Zaradi preprostosti in uporabnosti skriptnih programskih jezikov razvijalci uporabljajo še `Shell` in `Perl`. Taki vtičniki so počasnejši, saj tolmač sproti prevaja vrstico po vrstico.



Slika 3.5: Prikaz delovanja vtičnikov [8]

Uradni vtičniki so javno dostopni na uradni spletni strani nadzornega sistema Nagios. Paket vsebuje okoli petdeset vtičnikov. Obstajajo pa tudi neuradni vtičniki, ki jih razvijajo uporabniki nadzornega sistema po celem svetu. Število takšnih vtičnikov je bistveno večje.

3.1.4 Dodatki za Nagios

Dodatki za Nagios (ang. `Nagios Addons` [23]) razširjajo funkcionalnost nadzornega sistema, obstajajo pa tudi taki, ki omogočajo združitev nadzornega sistem z drugimi sistemi. Dodatki nam omogočajo, da si nadzorni sistem prilagodimo svojim potrebam.

Nekaj najpogostejših dodatkov omogoča:

- urejanje nastavitvenih datotek preko spletnega vmesnika,
- preverjanje stanja omrežnih naprav in storitev,
- izvajanje pasivnih poizvedb na oddaljenih napravah,
- poenostavljanje/razširjanje logike sporočanja ter
- shranjevanje podatkov v podatkovno bazo.

Na uradni spletni strani nadzornega sistema Nagios je možno dostopati do najpogostejših in najpopularnejših dodatkov. Ravno tako kot pri vtičnikih, pa obstaja tudi zelo veliko število neuradnih dodatkov. Te si lahko prenesemo iz spletne strani [24].

3.2 Sistem za pregled računalniške opreme Open-AudIT

3.2.1 Open-AudIT skozi zgodovino

Leta 1999 je Mark Unwin napisal ukazno datoteko (ang. `script`), ki je omogočala pregled računalnikov v lokalnem omrežju. Ukazno datoteko je dopolnjeval in kmalu je njegovo delo preraslo v projekt Open-AudIT, ki je temeljil na spletni aplikaciji.

Leta 2005 se je odločil, da projekt preda odprtokodni skupnosti v skladu z licenco GNU GPL verzija 2.

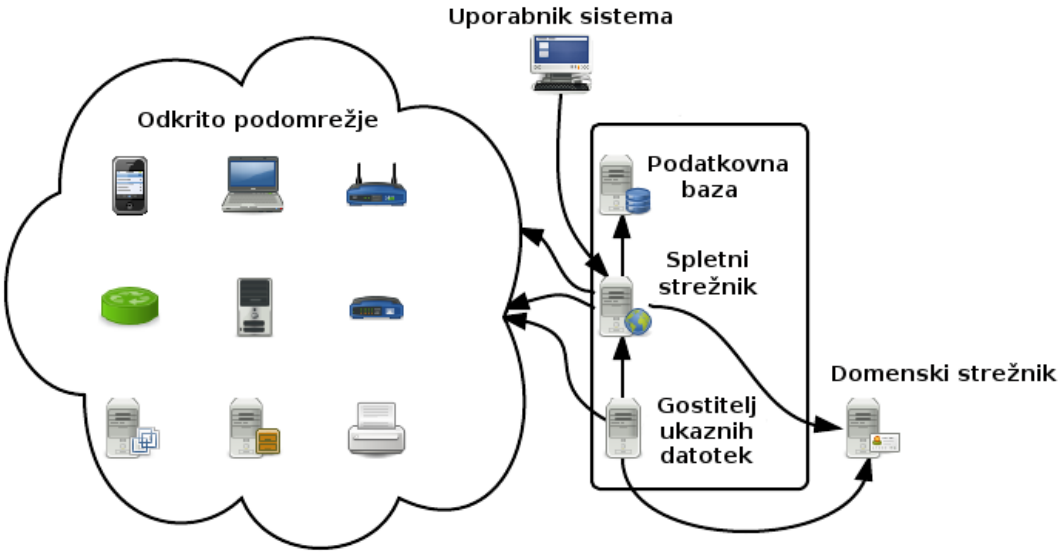
V letu 2008 je prišlo do izdaje razširjene verzije, ki je posodobila velik del spletne aplikacije in v veliki meri izboljšala funkcionalnost spletne aplikacije. Licenco GNU GPL verzija 2 je zamenjala novejša GNU GPL verzija 3.

Leta 2013 je Opmantek [25] pridobil pravice do intelektualne lastnine Open-AudIT z obljubo, da bo projekt ostal prost in odprtokoden. Do danes je temu tako, trenutno pa je možno prenesti poleg ostalih starejših tudi zadnjo verzijo 1.4.

3.2.2 Arhitektura sistema Open-AudIT

Open-AudIT [15] sestavljajo tri osnovne komponente: podatkovna baza, spletni strežnik in gostitelj pregledovalnih ukaznih datotek (slika 3.6). V manjših organizacijah je tipično, da so vse tri osnovne komponente na istem strežniku, toda v primeru velikih organizacij z velikim številom omrežnih naprav, je potrebno vsako komponento namestiti na svojega gostitelja. S tem razbremenimo posamezno komponento. Gostitelj lahko temelji na GNU/Linux ali Windows operacijskem sistemu.

Trenutno sistem podpira namestitev podatkovne baze MySQL. Open-AudIT temelji na PHP spletnem aplikacijskem okvirju (ang. `PHP web application framework`) CodeIgniter [26]. Ta okvir uporablja klasični poizvedbeni jezik SQL, zato ni težav pri uporabi kakšne druge podatkovne baze.



Slika 3.6: Arhitektura sistema Open-AudIT [15]

Bistvo sistema je spletni strežnik, saj se na njem izvaja celotno upravljanje. Pri izbiri spletnega strežnika je potrebno upoštevati podporo skriptnemu jeziku PHP (avtor sistema svetuje `Apache`). Glavni del logike sistema je na spletnem strežniku, nekaj malega v ukaznih datotekah za pregled. Poleg obdelave podatkov in prikaza spletnih strani, spletni strežnik poizveduje še po napravah s pomočjo ukaznih datotek.

Gostitelj pregledovalnih datotek je omrežna naprava, na kateri so nameščene ukazne datoteke, namenjene pregledovanju. To je lahko sam spletni strežnik, ali pa kakšna druga omrežna naprava, ki vrača rezultate poizvedb spletnemu strežniku. Zagon teh datotek uredijo sistemske aplikacije, namenjene nadzoru izvajanja programa v ozadju (ang. `Job scheduler`).

3.2.3 Vnos podatkov v sistem Open-AudIT

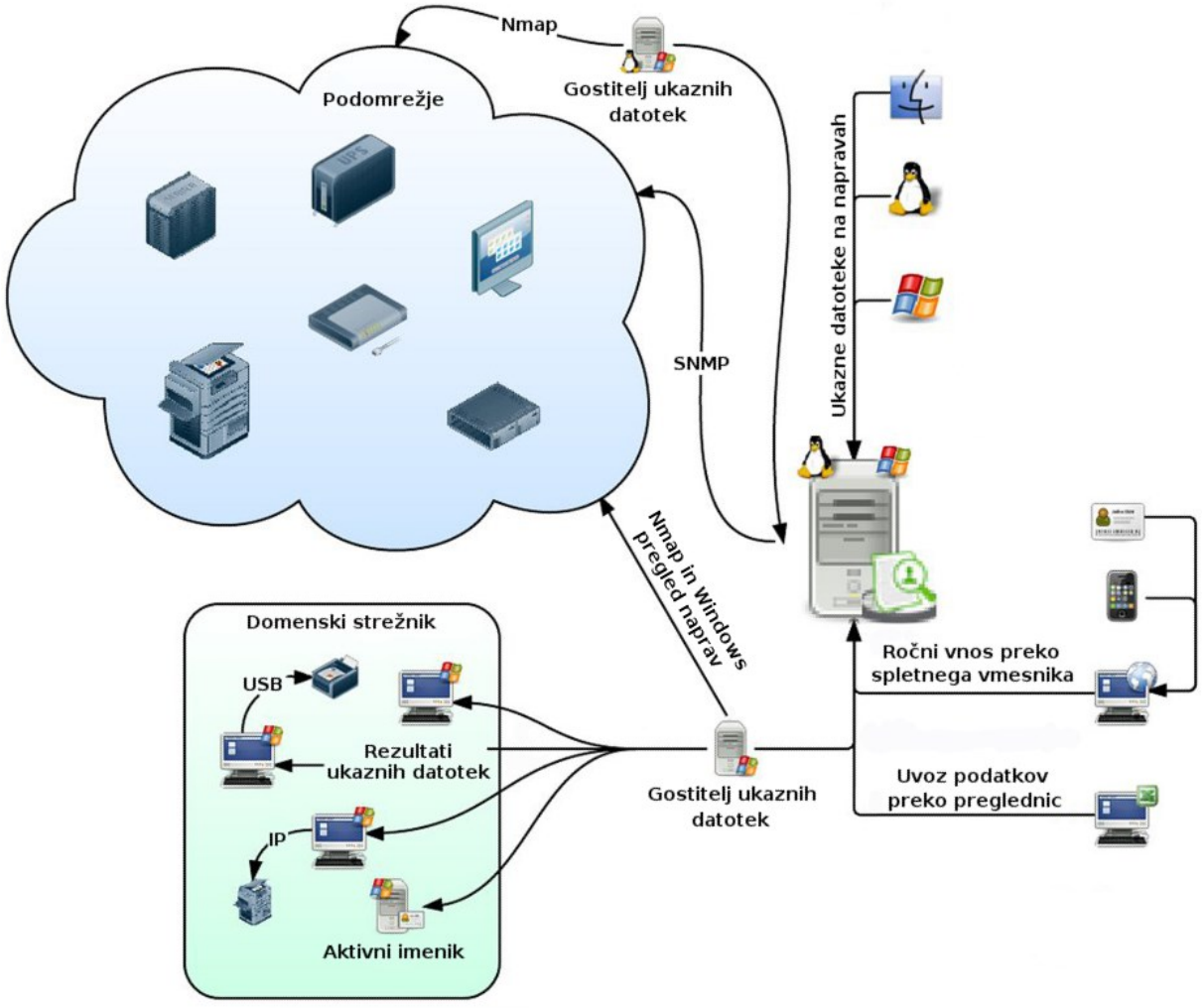
Sistem Open-AudIT lahko prejema podatke iz različnih virov (slika 3.7). Glavni vir podatkov so priložene ukazne datoteke, ki so namenjene pregledovanju operacijskih sistemov (Windows, GNU/Linux in OSX) in jih zaganjamo iz ukazne vrstice teh sistemov. Z zagonom ukaznih datotek na operacijskih sistemih, zberemo podatke o pregledovani napravi in jih posredujemo sistemu Open-AudIT.

Uporaben vir podatkov so tudi podatki, ki so generirani z ukaznimi datotekami na podlagi zajetih podatkov programske opreme `NMAP` [16]. Ukazne datoteke lahko zaženemo preko spletnega vmesnika ali ukazne vrstice. Ko `NMAP` ugotovi, da se na podomrežju nahaja naprava, se izvede `SNMP` pregled ciljne naprave. Zbrani podatki s tem načinom pregleda so

zadovoljivi, če so ciljne naprave usmerjevalniki, stikala ali kakšna druga omrežna naprava. V ostalih primerih je priporočljivo uporabljati drug način pregleda omrežnih naprav.

V primeru uporabe domenskega strežnika lahko naredimo pregled celotne domene ter preko aktivnega imenika dobimo podatke o napravah.

Spletni vmesnik sistema Open-AudIT omogoča ročni vnos podatkov posamično ali preko uvoza po predlogah že narejenih preglednic. To nam pride prav, ko se naprave v omrežju ne odzovejo na omrežni zajem, naprave niso priklopljene v omrežje ali naprave niso omrežne naprave.



Slika 3.7: Diagram pregledovanja naprav in vnosa podatkov [15]

Poglavje 4

Razvoj vtičnika

4.1 Delovanje vtičnikov

Sistem Nagios upravlja z vtičniki tako, da jih sproža iz ukazne vrstice. Ti vtičniki so lahko na nadzorovani napravi ali sistemu za nadzor. Sproža jih na način, da jim posreduje določene parametre, ki jih definiramo v konfiguracijski datoteki. Paziti moramo, da te parametre pozna tudi vtičnik. Vtičniku sistem tako posreduje podatke, na podlagi katerih vrne izhodne podatke. Vrnjeni podatki tako vsebujejo izhodno kodo in opis stanja, v katerem se nahaja nadzirana naprava ali storitev.

Iz tabele 4.1 lahko razberemo specifična pravila, ki veljajo pri razvoju.

Izhodna koda	Stanje	Opis
0	OK	Deluje pravilno
1	WARNING	Deluje, vendar potrebuje pozornost
2	CRITICAL	Ne deluje pravilno ali potrebuje pozornost
3	UNKNOWN	Vtičnik ni uspel preveriti stanja naprave ali storitve

Tabela 4.1: Pomen izhodnih kod vtičnikov [27]

Opis stanja je omejen z dolžino do 80 znakov, ki morajo biti zapisani v eni vrstici. Nagios opisa stanja ne spreminja in ga izpiše natanko tako, kot ga definiramo z vtičnikom. Po navadi se uporablja naslednja notacija:

```
STANJE VTIČNIKA - opis stanja
```

Priporočljivo je, da nam opis stanja vrne čim več informacij, s katerimi kasneje lažje odpravimo napake delovanja omrežnih naprav in storitev.

Pomembno je paziti tudi na to, da se sistemske ukaze kliče s polno potjo do programa. Tukaj je le nekaj osnovnih pravil, vsa ostala pravila lahko najdemo na [28].

4.2 Vtičnik za nadzor delovanja programskega čezmernega polja samostojnih diskov

4.2.1 Posnetek stanja

V podjetju uporabljamo na vseh ključnih napravah (strežniki in ključne delovne postaje) `RAID` polje diskov. Na uradni spletni strani Nagios vtičnikov [21] smo našli vtičnike, ki so zadovoljili potrebe po nadzoru delovanja `RAID` polja na večini naših naprav. Težavo smo imeli na namenski napravi. Ta naprava predstavlja naš privzeti prehod za dostop do spleta, požarni zid in poštni strežnik, kjer deluje programsko `RAID` polje. Naprava v podjetju opravlja zelo pomembno funkcijo, zato je treba zagotoviti njeno nemoteno delovanje. Za to napravo veljajo zelo visoki varnostni kriteriji. Iz spleta je naprava vidna le za potrebe delovanja poštnega strežnika, in oddaljenega dostopa. Varnostni ukrepi so za lokalno omrežje nekoliko drugačni, saj se dopušča dostop do `SSH` lupine iz lokalnega omrežja za določene naprave. Od naprave proti lokalnemu omrežju imamo za določene naprave dovoljen oddaljen dostop.

4.2.2 Kriteriji za razvoj vtičnika

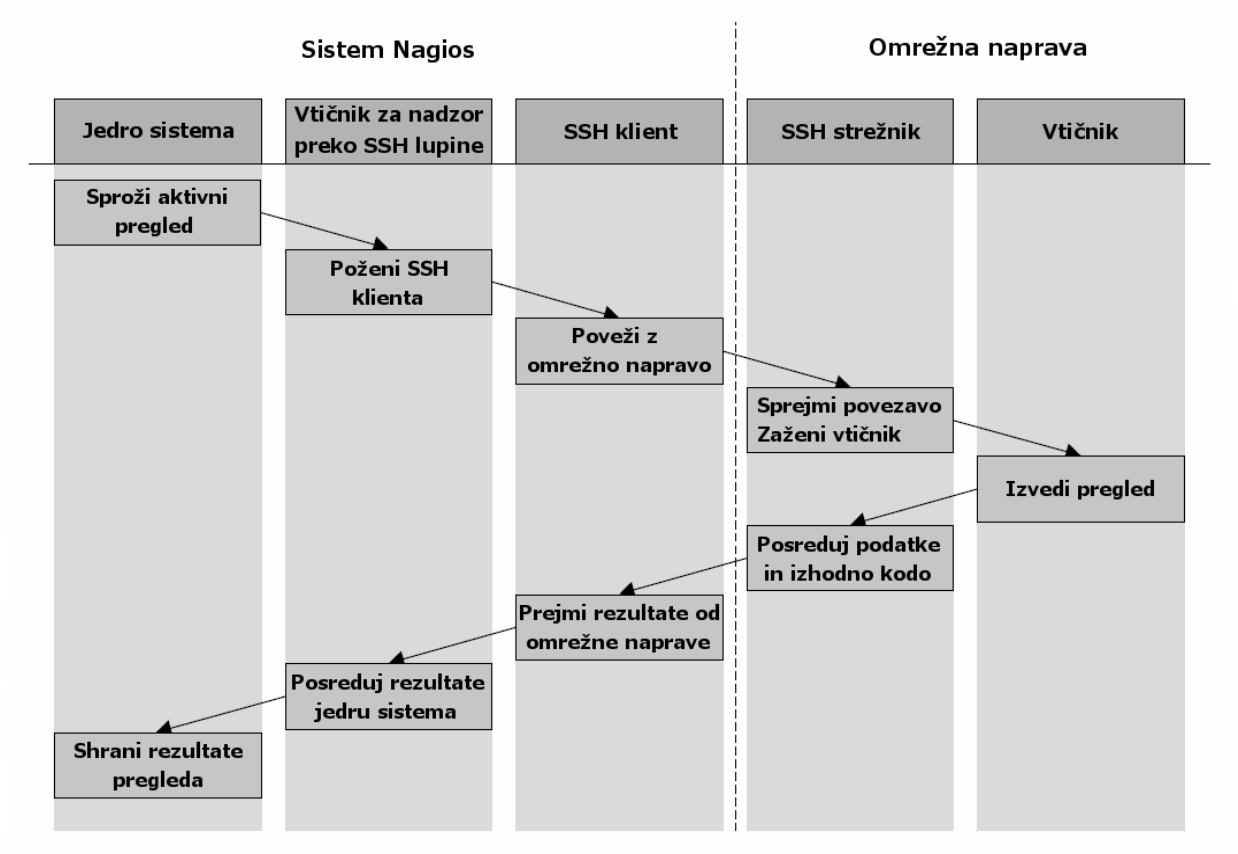
Varnostni kriteriji, ki so vplivali na razvoj vtičnika:

- nadzor ne spreminja lokalne varnostne politike dostopa do naprave,
- vtičnik mora delovati s pravicami uporabnika in
- delovati mora brez dodatnih namestitev programske opreme.

Na podlagi varnostnih kriterijev smo se odločili, da uporabimo aktivno metodo nadzora preko `SSH` lupine. Privzeta namestitev paketa `Nagios Plugins` vsebuje vtičnik, ki nam omogoča, da preko `SSH` lupine zaganjamo druge vtičnike. Ti vtičniki so nameščeni na nadzorovani napravi in z njihovo pomočjo nadziramo delovanje lokalne storitve. Njegovo delovanje lepo opisuje slika 4.1.

Pri naslednjem kriteriju je bilo potrebno preučiti možnosti, da bo vtičnik uporabljal sistemska orodja, za katera ne potrebuje skrbniških pravic. Orodje, ki ga potrebujemo pri analizi delovanja programskega čezmernega polja samostojnih diskov `grep`, ne potrebuje skrbniških pravic. Tudi proces, ki ga pregledujemo, dovoli uporabniku, da preveri status delovanja (`/proc/mdstat`).

Zadnji kriterij je odločal o tem, kateri programski jezik bomo uporabili za izvedbo vtičnika. Zaradi varnosti na naši napravi ni bilo podpore večini programskih jezikov. Ker gre pri nadzoru delovanja programskega RAID polja predvsem za manipulacijo s tekstom, smo se odločili da uporabimo awk (ang. interpreted programming language designed for text processing) in shell (Bash) ukazno datoteko za tolmačenje ukaznih vrstic.



Slika 4.1: Proces delovanja vtičnika za nadzor preko SSH lupine [27]

4.2.3 Razvoj vtičnika

Za dober nadzor storitve je potrebno, da zadovoljimo vsem pogojem razvoja vtičnika. V poglavju delovanje vtičnikov so navedena stanja, ki jih moramo upoštevati. Stanja dobimo iz statusa storitve. Poleg dobrega razumevanja razvoja vtičnika, moramo dobro razumeti tudi delovanje storitve, ki jo nadzorujemo.

Za prikaz statusa delovanja programskega RAID polja izvedemo ukaz cat /proc/mdstat. To so tudi izhodni podatki, s pomočjo katerih moramo določiti stanja delovanja storitve.

Ko RAID polje deluje brez težav sem nam prikažejo naslednji podatki:

```
Personalities : [raid1]
read_ahead 1024 sectors
md0 : active raid1 hdb1[1] hda1[0]
10241280 blocks [2/2] [UU]

md1 : active raid1 hdb2[1] hda2[0]
27655808 blocks [2/2] [UU]

unused devices: <none>
```

V našem primeru vidimo, da imamo na napravi fizično vgrajena dva trda diska hda in hdb, ki sta za potrebe delovanja RAID polja razdeljena na dve diskovni particiji. Disk hda na hda1 in hda2, ter hdb na hdb1 in hdb2. Vidimo lahko tudi, da so te diskovne particije povezane po sistemu RAID1 v polje md0 (hdb1 in hda1) in md1 (hdb2 in hda2). RAID1 nam omogoča zrcaljenje (ang. mirroring), kar pomeni, da se podatki zapisujejo na disk 1, ki je lahko hda ali hdb in njihova zrcalna slika se hkrati zapisuje tudi na disk 2 (to pa je ravno nasprotno, se pravi hdb ali hda). S tem se bistveno poveča varnost delovanja naprave, saj je trdi disk komponenta naprave, ki je najbolj podvržena okvaram. Podatek, ki je zapisan med oglatimi oklepaji ([UU]) nam pove, da oba diska, ki sestavljata polje md0 ali md1, delujeta brez napak.

V primeru napake delovanja programskega RAID polja ali odpovedi enega izmed diskov, bi se pri izpisu statusa delovanja prikazali naslednji podatki:

```
Personalities : [raid1]
read_ahead 1024 sectors
md0 : active raid1 hdb1[1]
10241280 blocks [2/1] [U_]

md1 : active raid1 hdb2[1]
27655808 blocks [2/1] [U_]

unused devices: <none>
```

Iz prikazanih podatkov je razvidno, da RAID polje ne deluje pravilno. Vidimo, da je v RAID polju aktiven samo disk hdb. To je tudi lepo razvidno iz podatka, ki se nahaja med oglatimi oklepaji ([U_]). Ta podatek je ključen pri določanju stanja OK in CRITICAL.

Programsko RAID polje se lahko nahaja še v stanju obnove in sinhronizacije. Namen postopka obnove je, da se poškodovani (napaka na RAID polju) ali novi disk prepiše s podatki iz delujočega polja, postopek sinhronizacije pa zagotavlja, da so podatki v RAID polju sinhronizirani.

V primeru obnove se poleg ostalih podatkov pri izpisu statusa delovanja prikažejo še naslednji podatki:

```
[===>.................]      recovery   =   15.2%   (4223616/27655808)
finish=61.8min speed=6309K/sec
```

V primeru sinhronizacije pa so poleg ostalih podatkov še naslednji:

```
[======>..............]  resync = 34.5% (9564416/27655808)
finish=25.2min speed=11927K/sec
```

Ko je programsko RAID polje v stanju obnove ali sinhronizacije, je priporočljivo, da z morebitno konfiguracijo sistema ali ponovnega zagona naprave počakamo, da se stanje obnove ali sinhronizacije uspešno zaključi. V ta namen smo se odločili, da, ko se naprava nahaja v tem stanju, vtičnik vrne stanje WARNING.

Če iz kakršnegakoli razloga ne moremo preveriti statusa delovanja programskega RAID polja z ukazom cat /proc/mdstat, nam vtičnik vrne stanje UNKNOWN.

Na podlagi predstavljenih stanj smo razvili vtičnik. Koda ukazne datoteke, ki nam zagotavlja nadzor delovanja vtičnika, je naslednja:

```
#!/bin/bash
#

# Pot do programa grep
EGREP='/bin/grep -E'
# Status delovanja programskega RAID polja
MD_PATH=/proc/mdstat

# Izhodne kode za sistem Nagios
STATUS_OK=0
STATUS_WARNING=1
STATUS_CRITICAL=2
STATUS_UNKNOWN=3

# Preverimo ce je status dosegljiv
```

```
if [[ -f $MD_PATH ]]; then
  # Stevilo md raid naprav
  MD_DEVICES=`$EGREP md $MD_PATH -c`

  # Stevilo poskodovanih tabel
  MD_DEGRADED=`$EGREP 'U_|_U' $MD_PATH -c`

  # V primeru okrevanje tabele, pokazi odstotek okrevanja
  MD_RECOVERING=`$EGREP recovery $MD_PATH | awk '{print $4}'`

  #V primeru sinhronizacije, pokazi odstotek sinhronizacije
  MD_RESYNCING=`$EGREP resync $MD_PATH | awk '{print $4}'`

  # WARNING - Ponovna sinhronizacija tabele
  if [[ $MD_RESYNCING ]]; then
     OUTPUT="WARNING - Resyncing array. Status: $MD_RESYNCING"
     STATUS=$STATUS_WARNING

  # WARNING - Tabela je v stanju obnove
  elif [[ $MD_RECOVERING ]]; then
     OUTPUT="WARNING - Degraded $MD_DEGRADED arrays, recovering one.
Status: $MD_RECOVERING"
     STATUS=$STATUS_WARNING

  # OK - RAID polje deluje brez tezav
  elif [[ $MD_DEGRADED == '0' ]]; then
     OUTPUT="OK - All $MD_DEVICES md devices are OK."
     STATUS=$STATUS_OK

  # CRITICAL - Resne tezave z RAID poljem
  else
     OUTPUT="CRITICAL - Degraded $MD_DEGRADED arrays. !!!"
     STATUS=$STATUS_CRITICAL

  fi
# UNKNOWN - Ni možno preverit statusa
else
OUTPUT="UNKNOWN - Something went wrong !"
STATUS=$STATUS_UNKNOWN

fi
```

```
# Status and quit
echo $OUTPUT
exit $STATUS
```

4.2.4 Preizkus delovanja vtičnika

Preizkus vtičnika za nadzor delovanja programskega RAID polja, smo opravili v testnem okolju. Tudi tokrat smo uporabili programsko opremo VMWare Workstation 10 [19]. Vzpostavili smo virtualni strežnik, ki je po konfiguraciji približno ustrezal namenski napravi, za katero je bil vtičnik razvit. Pomembno je bilo, da smo na virtualnem strežniku za namestitev uporabili dva virtualna diska. Da smo virtualna diska lahko medsebojno povezali po sistemu RAID1 (md0 in md1), smo ju morali pravilno razdeliti na diskovne particije.

Na tako pripravljenem virtualnem strežniku smo preizkusili možna scenarija:

- odpoved enega izmed delujočih virtualnih diskov (obe možni kombinaciji),
- odpoved particije trdega diska (več možnih kombinacij).

Ko smo vtičnik podrobno preizkusili in bili zadovoljni z njegovim delovanjem, smo ga namestili na namensko napravo in integrirali v obstoječ sistem za nadzor.

V obdobju uporabe vtičnika za nadzor namenske naprave se je izkazalo, da vtičnik dobro služi svojemu namenu. V času, ko nadzora nismo imeli, je bilo potrebno spremljati status delovanja ročno preko SSH lupine. To delo je bilo zamudno, hkrati pa tudi zelo neučinkovito, saj se je nadzor izvajal zelo poredko. Včasih nismo opravili nadzora tudi po več dni, saj je bilo potrebno zagotavljati delovanje tudi drugim omrežnim napravam in storitvam v podjetju.

Z integracijo vtičnika v obstoječi sistem za nadzor računalniške opreme nismo pridobili le na času. Bistveno je, da smo pridobili tudi nadzor, ki se vrši ves čas delovanja naprave.

Poglavje 5

Zaključek

Na trgu smo našli veliko brezplačnih sistemov namenjenih nadzoru in pregledu računalniške opreme. Zaradi tega smo morali v podjetju temeljito preveriti potrebe po takem sistemu in se odločiti za najbolj ustreznega.

Sistema, ki bi omogočal oboje hkrati, nismo našli, zato smo se odločili za dva ločena sistema, sistem za nadzor in sistem za pregled. Pri izbiri smo imeli težko nalogo, saj imajo odprtokodni (brezplačni) sistemi na tržišču zelo podobne funkcionalnosti. Na podlagi testiranja izbranih sistemov testnem okolju ter primerjalnih kriterijev smo se odločili za dva najprimernejša. Po tej fazi testiranja in izbora smo ju vključili v produkcijsko okolje. Pri tem se je izkazalo, da je bilo potrebno razviti vtičnik, ki bi omogočal nadzor delovanja programskega `RAID` polja na namenski napravi in zagotovil njegovo delovanje. Namenska naprava ima zelo visoke varnostne kriterije, kar je močno vplivalo na razvoj vtičnika. Za razvoj vtičnika smo tako uporabili `awk` in `bash`.

Odločitev za uvedbo sistemov za nadzor in pregled se je izkazala kot zelo dobra in zanesljiva, saj po uvedbi sistemov odprava napak na omrežnih napravah in storitvah poteka bistveno hitreje. Sistema sta primerna za mala in srednje velika podjetja, ki si komercialne programske opreme za namen nadzora in pregleda ne morejo privoščiti. Prav tako se je potrebno zavedati, da je treba nadzoru in pregledu dati zelo velik pomen, saj bi izpad ali okvara ključnih naprav v podjetju v najslabšem primeru lahko ogrozila njegovo delovanje. Z integracijo sistema smo več kot zadovoljni, saj sta tako Nagios kot Open-Audit zadovoljila vsa naša pričakovanja.

V našem podjetju imamo tudi omrežne naprave, ki so na oddaljenih lokacijah (prenosni računalniki, tiskalniki, itd.) in nimajo dostopa do lokalnega omrežja podjetja. Da bi bil nadzor in pregled popoln, bi bilo potrebno v obstoječ sistem povezati tudi te naprave. To bi lahko rešili na način, da bi jih v omrežje povezali z usmerjevalniki preko `VPN` (ang. `virtual private network`) povezav. S tem bi poleg nadzora in pregleda naprav njihovim uporabnikom nudili še pomoč na daljavo.

Literatura

[1] Zentyal, the Linux Small Business Server. [Online]. Dosegljivo: http://www.zentyal.org/.

[2] ReactOS Project. [Online]. Dosegljivo: https://www.reactos.org.

[3] Samba - Opening Windows to a Wider World. [Online]. Dosegljivo: http://www.samba.org/.

[4] Nagios vs Cacti vs Zabbix vs Zenoss. [Online]. Dosegljivo: http://www.serverfocus.org/nagios-vs-cacti-vs-zabbix-vs-zenoss.

[5] LinuxQuestions.org. [Online]. Dosegljivo: http://www.linuxquestions.org/questions/linux-enterprise-47/nagios-vs-zabbix-216790/page2.html.

[6] Poslovna Linux Konferenca. [Online]. Dosegljivo: http://www.linux-konferenca.org/.

[7] Konferenca Infosek. [Online]. Dosegljivo: http://www.infosek.net/.

[8] Nagios Documentation. [Online]. Dosegljivo: http://library.nagios.com/library/products/nagioscore/manuals/.

[9] Zabbix Documentation. [Online]. Dosegljivo: http://www.zabbix.com/documentation.php.

[10] Intelligent Platform Management Interface. [Online]. Dosegljivo: http://www.intel.com/content/www/us/en/servers/ipmi/ipmi-home.html.

[11] Java Management Extensions (JMX) Technology. [Online]. Dosegljivo: http://www.oracle.com/technetwork/java/javase/tech/javamanagement-140525.html.

[12] XMPP Standards Foundations. [Online]. Dosegljivo: http://xmpp.org/.

[13] Zenoss Documentation. [Online]. Dosegljivo: http://www.zenoss.com/resources/documentation.

[14] Zope. [Online]. Dosegljivo: http://www.zope.org/.

[15] Open-AudIT Documentation. [Online]. Dosegljivo: https://community.opmantek.com/display/OA/Home.

[16] NMAP. [Online]. Dosegljivo: http://nmap.org/.

[17] OCS Inventory NG Documentation. [Online]. Dosegljivo: http://wiki.ocsinventory-ng.org/index.php/Documentation:Main.

[18] OCS Inventory NG Agents. [Online]. Dosegljivo: http://www.ocsinventory-ng.org/en/download/download-agent.html.

[19] VMWare Workststion. [Online]. Dosegljivo: https://www.vmware.com/products/workstation/.

[20] vSphere Hypervisor. [Online]. Dosegljivo: https://www.vmware.com/products/vsphere-hypervisor/.

[21] Nagios Plugins. [Online]. Dosegljivo: http://www.nagios.org/download/plugins.

[22] NetSaint. [Online]. Dosegljivo: http://netsaint.sourceforge.net/.

[23] Nagios Addon Projects. [Online]. Dosegljivo: http://www.nagios.org/download/addons.

[24] Nagios Plugins. [Online]. Dosegljivo: http://exchange.nagios.org/directory/Plugins.

[25] Opmantek. [Online]. Dosegljivo: https://opmantek.com/.

[26] CodeIgniter. [Online]. Dosegljivo: https://ellislab.com/codeigniter.

[27] W. Kocjan, *Learning Nagios 4*, Birmingham, UK: Packet Publising Ltd, 2014.

[28] Nagios Plugin Development Guidelines [Online]. Dosegljivo: https://nagios-plugins.org/doc/guidelines.html.

[29] D. Pahor, M. Drobnič, Leksikon računalništva in informatike, Založba Pasadena, 2002.

[30] Islovar. [Online]. Dosegljivo: http://www.islovar.org.